INFORMATION SOCIETY TECHNOLOGIES
(IST)
PROGRAMME

Information Society
Technologies

OpenMolGRID

Data Warehouse Software Specification

Contract Reference:	**IST-2001-37238**
Document identifier:	**OpenMolGRID-1-D1.1a-0101-2-1-DataWarehouseSpec**
Date:	**07/01/2004**
Work package:	**WP1: Grid Data Warehousing of Molecular Structure – Property (Activity) Information**
Partner:	**UT, UU, Negri, FZJ, CGX**
Lead Partner:	**UU**
Document status:	**APPROVED**
Classification:	**PUBLIC**
Deliverable identifier:	**D1.1a**

Abstract: This document describes the aspects of the data warehouse software specification as a necessary requirement for its implementation. It contains details relating to constraints placed on the implementation according to user requirements and describes data types for the logical model and some simple transformations.

Delivery Slip

	Name	Partner	Date
From	Damian McCourt	UU	15/09/03
Verified by	WPM's	All	13/10/03
Approved by	G.H.F.Diercksen (TC) R.Ferenczi (QE)	OMC CGX	04/11/03 03/11/03

Document Log

Issue	Date	Comment	Author
0-0	03/02/03	First Version	Damian McCourt Jean Jing Werner Dubitzky
0-1	27/02/03	Second Stable Version; version submitted in second progress and management report	Damian McCourt
0-2	04/06/03	Draft Version to remove some information and add references to this	Damian McCourt
1-0	03/09/03	Complete document redraft	Damian McCourt
2-0	15/09/03	Submitted for Authorisation	Damian McCourt
2-0	04/11/03	Document Authorised	Damian McCourt
2-1	07/01/04	Updated due to the change of the document template (version1.3)	Jean Jing

Document Change Record

Issue	Item	Reason for Change
0-1	Expanded XML output To make the specification complete Added Section 3 To describe the data sources required	Refers to Original Submission to EC
0-2	Removed section 3 into separate documents Several deliverables required the same descriptions so new documents were developed as reference documents Grammatical changes made in accordance with Partner reviews	Refers to old version 2.0 before resubmission to EC

1-0	In order to make the functionality of the data warehouse more clear it was decided to break the document into several separate deliverables and to redraft what currently exists Added Section 3 to describe characteristics of the data warehouse: There was a need to describe the expected size, location, usage, and data volumes expected by the data warehouse as well as the need to describe the frequency of updates Removed Sections 5.1, 5.2 and 5.3: These relate to integration with UNICORE and were removed to deliverable D1.1f Section 4 (previously section 3) : Reworded previous documents, redrew architecture diagram to be more explicit and added details relating to automated data retrieval Added Section 5 : Required to describe the metadata needs of the warehouse Reworded Section 6 and 7: To make reference to the Project Quality Plan Added Logical Model: To describe how data will be represented in MOLDW Added Transformations: To describe the transformations currently envisaged Removed Use Cases : They were no longer useful and did not expand on those present in other deliverables	Changes made to address reviewers comments
0-1	Expanded XML output To make the specification complete Added Section 3 To describe the data sources required	Refers to Original Submission to EC
2-0	Document status changed to approved	WPMs, TC and QE Approved Document
2-1	The format of the head file is changed	The standard template of the document is changed

Files

Files in this section relate to actual storage locations on the BSCW server located at https://hermes.chem.ut.ee/bscw/bscw.cgi. The URL below describes the location on BSCW from the root OpenMolGRID directory

Software Products	User files / URL
Word 2000/XP	OpenMolGRID/Workpackage 1/Deliverables/ OpenMolGRID-1-D1.1a-0101-2-1-DataWarehouseSpec

Project information

Project acronym:	OpenMolGRID
Project full title:	Open Computing GRID for Molecular Science and Engineering
Proposal/Contract no.:	IST-2001-37238
European Commission:	
Project Officer:	Annalisa BOGLIOLO
Address:	European Commission - DG Information Society F2 - Grids for Complex Problem Solving B-1049 Brussels Belgium
Office	BU31 4/79
Phone:	+32 2 295 8131
Fax:	+32 2 299 1749
E-mail	annalisa.bogliolo@cec.eu.int
Project Coordinator:	Mathilde ROMBERG
Address:	Forschungszentrum Jülich GmbH ZAM D-52425 Jülich Germany
Phone:	+49 2461 61 3703
Fax:	+49 2461 61 6656
E-mail	m.romberg@fz-juelich.de

Contents

1. Introduction

1.1. Purpose and Scope

The purpose of this document is to specify particular aspects of the OpenMolGRID data warehousing component. This component will be referred to as MOLDW throughout the remainder of this document.

This document is intended for other partners in the consortium who are both users and developers. It does not discuss the integration of individual data sources into MOLDW. These documents will be presented in their own right as deliverables and will be referred to later in this document.

1.2. Overview

In terms of data management and data mining, the MOLDW is an important component in the OpenMolGRID project. It will act as a source of information where data from various sources are consistently integrated and will be available across a Grid infrastructure. This document describes the general functionality of MOLDW and specifies certain components

The main purpose of MOLDW will be to respond to a user query. The expected input and output formats of each query must be specified. This makes it possible to integrate MOLDW with other technologies, such as UNICORE. In order to query MOLDW it must contain some data, meaning that the functionality to integrate data sources into MOLDW must be available. Each data source to be integrated into MOLDW will be analysed according to the information currently required by users as specified in deliverable D1.3 [1]. To accompany the analysis of these data sources, a logical data model will be developed. This data model is described in section 6 of this document.

1.3. Document Structure

In addition to this section the document contains the following sections:

- Section 2 – a general description of the requirements for MOLDW.
- Section 3 – characteristics of MOLDW
- Section 4 – a high level architecture for MOLDW
- Section 5 – a description of metadata for MOLDW
- Section 6 – a description of the logical model for MOLDW
- Section 7 – a description of the transformation required for MOLDW
- Section 8 – a description of the standards being adopted for MOLDW during development
- Section 9 – references

1.4. Terminology / Glossary

UT	University of Tartu, Estonia
UU	University of Ulster, UK
FZJ	Research Centre Juelich, Germany
Negri	Mario Negri Institute, Italy

CGX	ComGenex, Hungary
MOLDW	OpenMolGRID Data Warehouse
UNICORE	Uniform Interface to Computer Resources
TSI	Target System Interface
NJS	Network Job Supervisor
XML	Extensible Markup Language
SQL	Structured Query Language
Usite	UNICORE site
Vsite	UNICORE target system at a Usite
DBAT	Database Access Tool
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol
ms	Milliseconds
S	Seconds
M	Minutes
OGSA	Open Services Grid Architecture
ADME	Absorption, Distribution, Metabolism, Excretion
NTP	National Toxicity Program
CML	Chemical Markup Language

2. General Description

This section of the document will provide a general description of the requirements for MOLDW.

2.1. User Problem Statement

The aim of many people working in the chemical, biochemical and pharmaceutical industries is to address large-scale molecular design problems. Rather than synthesise new chemicals from scratch and test their properties, OpenMolGRID aims to predict properties based on information already known about other similar chemicals. If the properties of the chemical are desirable, then it is likely to be synthesised. This approach is computationally intensive and requires large volumes of data from varying sources. However it is the access to computing resources that limits the process and not the data volumes involved. The large data volumes come from the fact that there may be tens of thousands of chemicals in one data set, but the total size of this data is not in the scale of terabytes or petabytes. Gigabytes are more realistic.

The data from the various sources is often presented in a different way. For example, one data source may present its information for toxicity dosages using grams per kilogram while another may use milligrams per kilogram. This makes it difficult to automate the prediction process. Data warehousing can provide a partial solution to this problem. MOLDW will help with the prediction process by providing relevant data in a form that is consistent and aims to enrich data by performing transformations, such as unit standardisation. Further details relating to the molecular design process can be found in D3.6 [2] and details relating to quantitative-structure/activity relationships can be found in D2.4a [3].

2.2. MOLDW Users

MOLDW will contain data useful for the molecular engineering process. It is assumed that the users of MOLDW will be from this or a similar background. Their expertise with software systems is not expected to be of a high level, although in many cases it may be. The technical details of MOLDW will be hidden from the user, since UNICORE provides seamless access to computing resources. A description of UNICORE can be found in [4].

2.3. User Objectives

The main objective of the users of MOLDW will be to gain access to integrated and consolidated data that originates from various disparate data sources. It is important that MOLDW contains data from relevant data sources that are of value not only within OpenMolGRID, but also to the wider community. This data will have a variety of uses, but the main one will be as a pre-cursor to the molecular modelling process carried out in workpackages 2 and 3. Some of the concepts inherent in data-warehousing methodologies are often pre-cursors to the data-mining process that is part of molecular modelling. A general description of data warehousing is discussed in D1.4e [5].

2.4. MOLDW Function

In order to meet the user's objectives, there are two main requirements that MOLDW must fulfil. Users must be able to access MOLDW and MOLDW must be loaded/ populated with data from external data sources. An external data source is a necessarily vague term to describe a source of data that exists outside the control of MOLDW.

The data sources that are currently being considered for integration into MOLDW are listed below:

1. Aquire (Ecotox)
2. Terretox (Ecotox)
3. National Toxicity Program or NTP

These data sources are described in D1.4a [6] and D1.4b [7] respectively. Relevant information from these sources must adhere to the logical data model outlined in section 6 of this document. An analysis of the information in each data source with the information required in the logical model is described in D1.1b [8], D1.1c [9] and D1.1d [10] respectively.

When data is available in MOLDW a mechanism to access this data must exist. The only access mechanism to MOLDW that will be developed during the OpenMolGRID project will be via the Grid, i.e. UNICORE. Integration with UNICORE will adhere to the general architectural approach being adopted by OpenMolGRID as outlined in D4.5a [4]. This access will be realised with the implementation of a Database Access Tool (DBAT) specific to MOLDW. A description of the general database access can be found in deliverable [11]. The description of the specific DBAT for MOLDW access is described in D4.1c [12]. To enable the use of MOLDW in the general user workflow, described in D4.5a [4], a user interface to database access must be developed. This is described in D4.1a [13].

In addition to the provision of data access, MOLDW must also provide access to metadata. Metadata will describe what data is contained in MOLDW and the metadata itself must be understandable by the users of MOLDW, i.e. molecular engineers. As metadata is essentially data, it can be accessed in the same way as “normal” or operational data. However, the separation of metadata from operational data at the architectural/ conceptual level brings about greater flexibility to the overall process, even if both are represented in the same form, e.g. in a relational database management system (DBMS). A description of the metadata to be contained in MOLDW is described in section 5 of this document.

Data warehouses in general require some form of administration [5]. Within OpenMolGRID, users will have restricted access to MOLDW. They will only access MOLDW using the read only UNICORE interface. Currently no administration will take place across the Grid. The data warehouse administrator will carry out administration locally. Inconsistency checks will require manual interaction. Some data may require manual insertion onto MOLDW, as they are “one-time-only” sources. This is envisaged during the initial population of MOLDW.

3. MOLDW Characteristics

This section of the document will describe some constraints that must be placed on MOLDW in relation to physical resources, performance requirements and user requirements.

3.1. Central Physical Storage or Repository

Fundamental to the data warehousing process as outlined in D1.4e [5] is the need for either a virtual or physical central information repository. Due to the nature of the data, (see D1.3 [1]), the potentially complex data transformations (see section 7) and the expected frequency of data requests (see section 3.5) within OpenMolGRID, a physical central repository will be developed. This will offer significant performance advantages when the alternative is considered. To integrate various data sources for every request received is unrealistic and extremely time-consuming making physical storage is more suitable. A virtual repository offers no advantages over a physical on in the context of OpenMolGRID. This may change in future as more data sources and types are added to MOLDW.

3.2. Physical Realisation

Data warehouses are often developed using standard DBMSs or with simple extensions to them. The physical implementation of MOLDW will be realised using PostgreSQL. This choice was made based on a comparison between MySQL and PostgreSQL as described in technical document DBMS Comparisons [14].

3.3. MOLDW Size

Currently there are three data sources that have been considered for integration into MOLDW. Their raw data sizes are summarised in ***Table 1***.

Table 1*:* Data volumes of individual data sources

Source	Raw Data Size
Terretox	46.1 MB
Acquire	76.4 MB
NTP	27.1 MB
Total	149.6MB

In total, this accumulates just less than 150 MB. Not all of the data contained within these sources are relevant for insertion into MOLDW.

While data warehousing results in the storage of largely redundant data, it would be reasonable to assume that this figure would increase after the data sources are processed. However, the data stored in these sources is already largely redundant. Taking this into account, it is unlikely that under the timescale of this project the data warehouse will be greater than 5GB. However, given that data sources may change or be added at any time this figure may increase, but will still be in the gigabyte range.

3.4. MOLDW usage

There are three main usages of the data contained within MOLDW. These are described below:

1. A user may access the data for the purposes of finding descriptors or experimental values for specific compounds. They may do this from one to one hundred times a day. Each query would return from zero to couple of hundred records consisting of structures, property and descriptor data.
2. A user may access MOLDW if they are trying to build a model. A typical model would require from 100 to 10000 compounds. The user would perform this type of query one to five times a day.
3. A user may access MOLDW if they are aiming to predict values in a large-scale environment. Most of the compounds available will be retrieved in one query. This type of usage would be typical of pharmaceutical companies. This type of query would be performed no more than once per day.

The combination of NTP and Exotox (Both Aquire and Terretox) will provide information for around 10000 chemicals.

3.5. Expected Requests

The usage by one client may vary from one to few hundred of queries per day. During the lifetime of the project it is expect that the number of clients will be about 5 in total. This means a maximum of 500 queries per day are expected during the project. This estimate is optimistic. The size of the queried data may vary depending on the nature of the query.

3.6. Query Response Data volume

The size of the data response is largely dependant on the type of query carried out. This directly affects the time to formulate the response. One molecular structure would require about 2-3KB. One property and or descriptor value for a structure would require less than 100 bytes. The full set of property and descriptor values would require about 50-80KB per structure. The average structure will require no more than 100KB per record (including compounds, properties and descriptors). These values are based on the fact that the data warehouse can be populated with the relevant data from public data sources. ***Table 2*** shows the estimated data volumes for queries from the data warehouse.

Table 2*:* Estimated size of query responses

Usage Type	Number of Compounds	Total Data Size
Experimental Data	1-100	0.1MB-10MB
Model Building	100-10000	10MB-1GB
Prediction	>10000	Size of Warehouse

Given the current data sources to be integrated into MOLDW, 5GB is the maximum estimated size.

3.7. Data Replication

Data replications can be viewed as the copying of data from its source to a destination or multiple destinations. Essentially it results in the same data (or partial versions) being available at multiple sites. This provides a distinct performance advantage, especially where there are huge volumes of data or high access demands (volumes and frequencies).

Given that the size of MOLDW is not expected to exceed 5GB and that the usage is not expected to be excessive, it is unlikely that there will be any performance issues resulting in the need for data replication. Within OpenMolGRID the data volumes are not the limiting factor of the process. The number of chemicals may be large, but their data volumes are not. It is the computational power required for the model building and prediction that restricts the process. However, with the further identification of data sources and a change in needs from the users, this may well change and data replication issues will have to be considered. For example, access demands may rise significantly if MOLDW is made available to the wider molecular engineering community.

3.8. Location

Given that a physical central repository is required, MOLDW must physically reside at a location within OpenMolGRID testbed. MOLDW will be located on a server based at the University of Ulster. The specification of the MOLDW server is shown ***Table 3***.

Table 3: Specification of MOLDW server machine

Attribute	Value
Machine	Sun LX 50
Operating System	Linux
Processor	2 x 1.4 GHz Intel III
RAM	2GB
Hard Drive	65GB

A UNICORE server will be installed on this machine to make MOLDW available as another resource within OpenMolGRID.

3.9. Frequency of updates

More and more information is being discovered every day meaning that information in any data source must be kept up-to-date. Within MOLDW it is likely that the underlying data sources will change with new data being generated, although this is not expected to be frequent. It is important that MOLDW avail of such information updates and should make it available to its users. The update frequency of these sources ultimately determines the update frequency of MOLDW. Some sources may update on a weekly basis, others monthly and still others not at all. The update of MOLDW will be considered on a source-by-source basis. However, the details of which sources must be updated, and when they are to be updated must be stored and triggered at the appropriate time. The update frequency applied to the data sources currently under consideration is described in ***Table 4***.

Table 4: Update Frequency of individual data sources

Source	Date Last Updated	Update Frequency
Aquire	05/05/2003	Quarterly
Terretox	05/05/2003	Quarterly
NTP	02/12/1996	Unknown – does not appear to be updated

Although the shortest update period described in ***Table 4*** is quarterly, it is envisaged that as MOLDW develops (with more sources being added) that an update frequency of once a month will be sufficient. However, depending on the dependencies between individual sources, not all information may require updating at the same time. Currently the NTP and Ecotox (both Aquire and Terretox) sources can be considered independent

Each data source increases in size as new data is added. To emphasise the fact that changes do not occur frequently the percentage change between the current and previous changes can be calculated. ***Table 5*** shows the percentage change in the current data sources that have regular updates. The percentage change is based solely on the difference in size of the data sources and no actual analysis of the internal data has taken place.

Table 5: Percentage change in data sources

Source	Current Size	Previous Size	% Change
Aquire	76.4MB	74.8MB	2.14%
Ecotox	46.1MB	42.9MB	7.46%

Given that the percentage change is relatively low for these sources, this emphasises the fact that data changes are infrequent and that an update frequency of once a month is sufficient.

3.10. MOLDW Update time

Given the current data sources to be integrated into the data warehouse it is possible to give a rough pessimistic guide as to how long the data update process will take. This estimate is based on pessimistic download rates and the read/write times of the largest file in each archive. This allows in part for the searching and logic that will be required for processing each source. ***Table 6*** shows the estimated download time for each individual data source.

Table 6: Estimated Download Time for Each Data Source

Archive	Download size (MB)	Estimated Time (m)
Aquire	10.20	13.60
Terretox	5.33	7.11
NTP	11.60	15.47
Total	27.13	36.17

These figures are based on the fact that there is a 34Mb connection from UU to external sites. The available bandwidth for the data warehousing process will be 34Mb minus what other users are using. It is expected that updates will take place in times of low network usage, but a pessimistic 100Kb connection has been assumed during this estimation.

Once each data source is downloaded they must be uncompressed and processed appropriately. In this estimation, the time taken to uncompress an archive has not been taken into account. The estimated processing time for each data source is described in ***Table 7***. These times are based on the time to read the largest file in a data archive, to write the file and to copy the file (read + write). The copy file has been added to allow in some way for the processing time of each file.

Table 7: Estimated Processing Time for Each Data Source

	Aquire	Terretox	NTP
Download Size	10.2MB	5.33MB	11.6MB
Archive type	Zip File (self-extracting)	Zip File (self-extracting)	Zip File
Uncompressed size	76.4MB	46.1MB	27.1MB
Total Files	53	43	2326
Files to process	40	33	2326

Largest File Size	47.7MB	10.6MB	33KB
Read Time (ms)	28000	6218	62
Write Time (ms)	34203	7688	64
Copy Time (ms)	62203	13906	126
Total (ms)	124406	27812	252
Total (s)	124.406	27.812	0.252
Total (m)	2.073	0.464	0.004
Processing Time (m)	82.937	15.297	9.769

While the worst-case scenario has been predicted in terms of the largest file size in each archive to be processed (i.e. it is assumed each file is the size of the largest file), it must be pointed out that this calculation does not take into account details associated with the logic surrounding the processing of these files or the data transformations that might take place. The pessimistic assumption will in some way make up for the time taken with transformations and logic. The total estimated processing time for data warehouse updates is shown in ***Table 8***.

Table 8: Total Time for MOLDW Update process

Property	Time (m)
Downloads	30.17
Aquire	82.937
Terretox	15.297
NTP	9.769
Total	144.18

Given these calculations data warehouse updates will take approximately 2.5 hours.

The effect of updates upon the system performance will be determined by how long the data warehouse is inaccessible and at what time these updates will take place. During data updates, the performance of MOLDW will degrade. To reduce the impact of this, updates will be carried out at a time when the system is regarded to be in a low use state. Currently this time will be scheduled for midnight CET at the end of each month. A downtime of 2.5 hours in one month is reasonable (0.5 %).

4. High Level Architecture

Given the high level description of the main functionality of MOLDW, it is possible to formulate a high-level architecture. This section of the document will describe the general architecture for MOLDW by mapping functionality to components and will highlight the need for certain interfaces.

4.1. Function to Component Analysis

The main function of MOLDW is to provide access to data. This will be provided by means of a Database Access Tool (DBAT). Within the OpenMolGRID project several DBATs will exist. DBAT_MOLDW will be the access tool for MOLDW. Access to metadata will also be provided through this tool. Other DBATs will access other databases. A description of the adaptor for general database access can be found in D4.1a [13]and D4.1c [11]and a description of the specific DBAT for MOLDW is described in D1.1f [12].

The main functionality for DBAT_MOLDW will be to connect to MOLDW, parse the user input (i.e. query) to an understandable format, execute the query and parse its response. The expected inputs and outputs of MOLDW are specified in D1.1f [12].

MOLDW must contain data. This means that a population process must take place based on the data sources that are to be integrated. A population tool will be responsible for downloading data, uncompressing it is appropriate and extracting the relevant data. Some data transformations may be necessary on certain parts of the data. The data retrieval and population process is described in section 4.4 of this document.

4.2. Interfaces

A general function to component analysis introduces the need for several interfaces. Each interface is discussed below.

4.2.1 JDBC

The main implementation language for the development of the tools to support MOLDW will be Java. Java Database Connectivity or JDBC provides a standardised interface to databases regardless of the type of the database. As long as there is a JDBC database driver for the DBMS, then the program code is independent. The database implementation can change with little or no impact on the developed software. This interface will be used by both the update mechanism and the access mechanism.

4.2.2 UNICORE – MOLDW Interface

DBAT_MOLDW will be used to provide access to MOLDW via UNICORE. It will have a command line interface. This tool will have the following syntax:

```
dbat_moldw infile outfile
```

In this sample infile relates to the query being sent to MOLDW and outfile relates to the response from MOLDW. The input and output of MOLDW is described in D1.1f [12]. UNICORE will initiate the execution of this tool using the syntax outlined above.

4.2.3 MOLDW – Data Source Interface

MOLDW must access data sources and process them appropriately. To complement the general architectural approach taken to integrate resources into UNICORE, a similar approach will be adopted in this interface. Further details of surrounding automated retrieval are discussed in section 4.4 of this document.

4.3. Initial Architecture

The function to component analysis, and the identification of interfaces allows an initial architecture diagram to be constructed. This is shown in Figure 1.

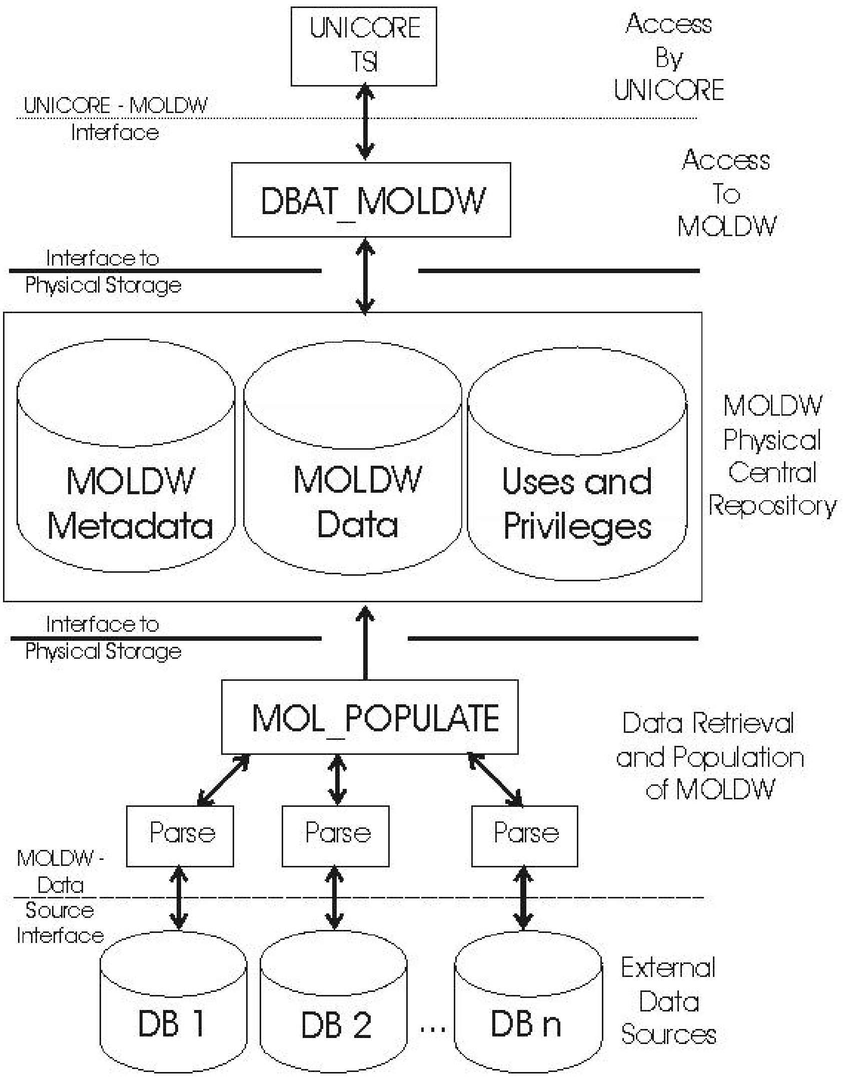

UNICORE TSI
Access By UNICORE
UNICORE - MOLDW Interface
DBAT_MOLDW
Access To MOLDW
Interface to Physical Storage
MOLDW Metadata
MOLDW Data
Uses and Privileges
MOLDW Physical Central Repository
Interface to Physical Storage
MOL_POPULATE
Data Retrieval and Population of MOLDW
Parse
Parse
Parse
MOLDW - Data Source Interface
DB 1
DB 2
...
DB n
External Data Sources

Figure 1: Architecture of MOLDW. Access from UNICORE will be provided with the development of DBAT_MOLDW. External data sources will be integrated with the development of a populate tool.

Figure 1 clearly shows that data sources lie outside MOLDW and are therefore considered external.

4.4. Data Retrieval and Processing

Currently the only data sources to be considered for integration into MOLDW are those that are publicly available for download and are archived. Given that the sources considered are publicly accessible it would be difficult to insist that a UNICORE TSI (Target System Interface) be placed on an external data source server. Therefore these sources will be accessed directly without the use of Grid. The automation of MOLDW population can therefore be considered as separate to data access.

In general data sources have different access mechanisms. Many sources make their data available in archived or compressed formats. This is currently the preferred format for MOLDW. This means no UNICORE TSI's are required on the target data source server and all data can be retrieved at once. Some sources limit the number of records that can be returned in one query, making archives more suitable. This gives MOLDW more control over its data, but this is also a potential risk. Data sources may change their formats or remove make their data no longer available to the general public. If either situation occurs MOLDW will have no choice but to adapt as necessary. External data sources must be monitored for changes.

MOLDW will require a data retrieval mechanism for each data source. Each source currently considered is available as an archive. To complement the general approach being adopted to access resources in UNICORE, MOLDW will access external data with the implementation of an Abstract Resource interface. Data sources are being made available through UNICORE by using Database Access Tools (DBATs). MOLDW itself will be accessed via one of these tools (DBAT_MOLDW [12]). An archive access DBAT tool will be developed so that the relevant archives can be retrieved and stored on the local system for further processing. Archives are usually available via HTTP or FTP. In order to use such a tool, certain pieces of information about the target source must be known. As a minimum the following information is required:

- The URL (HTTP or FTP) of the source
- The format of the file (e.g. zip, tar)
- How the file should be processed (i.e. what parsing mechanism to be used)

The URL must exactly identify which file is to be retrieved from the external system. The name of this file is specific to the individual data source. File archives, as their name suggests, contain data that is not in its raw format. They are usually compressed versions of the raw data. It must be known what compression format the archive has so that its raw data can be extracted. Common compression formats include tar and zip. Such archives can be decompressed via command line tools (e.g. tar, gunzip) available by default on operating systems such as Linux, under which MOLDW will operate (see ***Table 3***). Decompression will result in the raw information being made available for further processing.

Given that the data contained in a particular data source is specific to that source, the processing of raw data must be carried out on a source-by-source basis. This means a tool or parser must be developed for each source so that its data can be prepared before it is inserted into MOLDW. This parser will contain the necessary logic for the processing of the data source for which it is designed. Care must be taken during the design of such tools, as it is there is likely to be a substantial overlap in the required functionality.

Automated retrieval can be achieved by using operating system tools (such as cron) to automate when the update task should be started (and how often). This will initiate the download of each data source, decompress it and process it as appropriate.

5. Metadata

Metadata is generally accepted as data about data. Within MOLDW metadata will be used to describe the data that will be contained within it. In its simplest form this relates to a description of tables and fields.

Fundamental to the data warehousing process are data transformations. In order to aid subsequent processing of data for processes such as data mining, units will be standardised as a minimum. The original data prior to transformation will also be available should the user wish to view it.

Figure 2 shows how metadata will be represented within MOLDW.

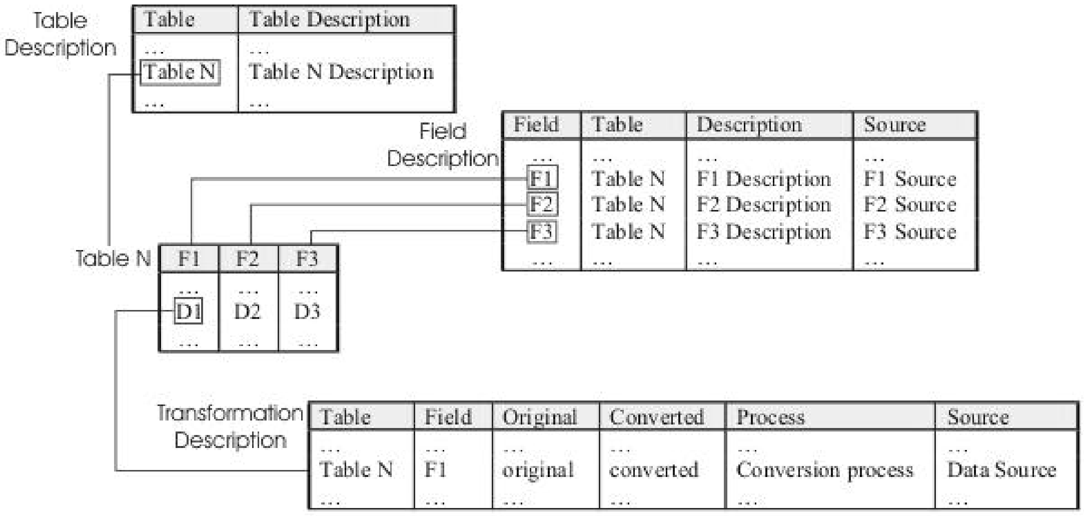


Table	Table Description
…	…
Table N	Table N Description
…	…

Field	Table	Description	Source
…	…	…	…
F1	Table N	F1 Description	F1 Source
F2	Table N	F2 Description	F2 Source
F3	Table N	F3 Description	F3 Source
…	…	…	…

F1	F2	F3
…	…	…
D1	D2	D3
…	…	…

Table	Field	Original	Converted	Process	Source
…	…	…	…	…	…
Table N	F1	original	converted	Conversion process	Data Source
…	…	…	…	…	…

Figure 2: Representation of metadata within MOLDW

In addition to this there will also be a higher-level description of the data contained within MOLDW to describe some general properties of MOLDW. An initial list of properties is shown in Table 9.

Table 9: Properties of MOLDW to be used for metadata purposes

Property	Description	Current Value (if applicable)
Name	The name of the resource	MOLDW - OpenMolGRID Data Warehouse
Creator	Who created the resource	OpenMolGRID
Subject	What the resource contains	Toxicity
Date	The date the resource was created	Date of First Upload – To be determined
Language	The language of the resource	English
Audience	Who the resource is intended for	Toxicologists, pharmacologists, molecular engineers, environmental science engineers,
Modified	The date the resource was last modified	Depends on

Access to metadata will occur using the same mechanism as access to operational or real data. This will be through the use of a DBAT (DBAT_MOLDW). While the access mechanism to metadata will be the same as that of operational data, conceptually it is different and so it is important to keep them separated at all levels, including the architectural. Metadata will also be stored in the central MOLDW repository.

6. Logical Model

This section of the document describes the logical model for MOLDW. Firstly entities will be identified, and then relationships determined resulting in the development of an Entity Relationship model. A basic assumption of the data at present is the fact that all data contained in MOLDW is public to everyone in the virtual organisation.

6.1. Entities

Having evaluated the user requirements for data there are several entities that can be identified. These are described in subsequent sections, but are summarised in Table 10.

Table 10:Entities and their description

Entity	Description
Chemical	Details associated with the chemical in question.
Structure	A structure associated with a chemical.
Descriptor Type	The type associated with the descriptor.
Descriptor	Items that have been calculated about a chemical whether public or private.
Experiment	An experiment that has been carried out.
Property Type	The type of property under consideration
Property	A property value of the chemical
Toxicity Measure	A toxicity measurement that has been performed using the chemical.
Source	The source of a particular experiment
Owner	The owner of a particular property, descriptor or structure
Software	The software used to generate a particular descriptor or structure

6.1.1 Chemical

The aim of this entity is to capture details relating to a particular chemical of interest that has been found in a data repository. A graphical representation of this entity is shown in ***Figure 3***.

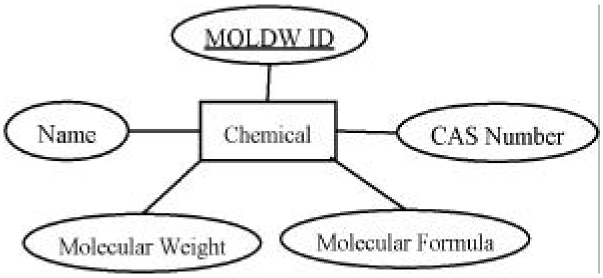


Figure 3: The Chemical Entity

Descriptions of the attributes for this entity are described in ***Table 11***.

Table 11: Attributes of the Chemical Entity

Attribute	Identifier	Description
MOLDW ID	Yes	The internal identifier associated with the chemical. An exact structure search must be carried out with the addition

		of each internal chemical to ensure that the current chemical is in fact a new chemical.
CAS Number	No	The Chemical Abstracts service number associated with the chemical. It is assumed that data in public repositories will always have a CAS number associated with them.
Name	No	The name associated with the chemical.
Molecular Formula	No	The molecular formula associated with the chemical
Molecular weight	No	The molecular weight of the chemical measured in AMU. If this field is unavailable it can be derived from the chemical formula. Details of this are described in section 7.1

6.1.2 Structure

Every chemical has an associated structure. This structure however can take many forms. It is possible that the structure may be in two dimensions or three dimensions. Depending on the conditions surrounding the structure generation, i.e. software used, there may be various forms of a structure resulting in a need for these details to be represented as an entity. The structure entity is shown in ***Figure 4***.

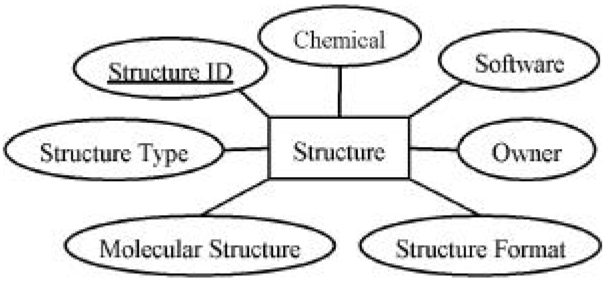


Figure 4: The Structure Entity

The attributes of this entity are shown in ***Table 12***.

Table 12: Attributes of the Structure Entity

Attribute	Identifier	Description
Structure ID	Yes	The ID associated with the type of the descriptor. There is no universal identifier for structure apart from the actual representation itself
Structure Type	No	The type of the structure. This may be either 2D or 3D.
Chemical	No	The chemical the structure relates to.
Molecular Structure	No	The actual structure represented in a textual format. Examples include a Mol File and CML file.
File format	No	The format of the file containing the structure
Software	No	The software that was used to generate the structure. If not immediately available, this attribute can be derived from if the format is a Mol File
Owner	No	The individual, laboratory or source of the structure

6.1.3 Descriptor Type

All descriptors calculated are of a particular type. To give flexibility to allow new types to be added, this entity has been identified. The attributes of this entity are shown in *Figure 5*.

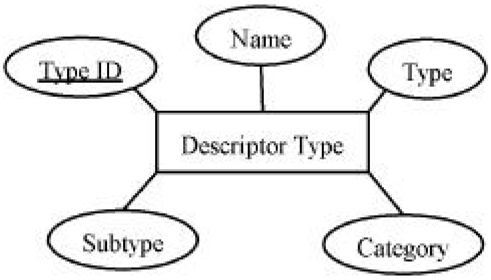


Figure 5: The Descriptor Type Entity

Descriptions of the attributes of this entity are described in ***Table 13***.

Table 13: Attributes of the Descriptor Type Entity

Attribute	Identifier	Description
Type ID	Yes	The ID associated with the type of the descriptor
Name	No	The name of the descriptor.
Type	No	The Type of the descriptor.
Subtype	No	The subtype of the descriptor.
Category	No	The category of the descriptor. This can be 1D, 2D, or 3D depending at which level of the structure representation it was obtained.

6.1.4 Descriptor

Descriptors describe particular aspects of the chemical that have been calculated using some piece of software. This kind of information is typically not found in public data repositories and must therefore be provided by other partners in the consortium. Typically these will be data sets that have been generated with some specific aim and care must be taken with the preparation of these data sets. The descriptor entity is described in ***Figure 6***

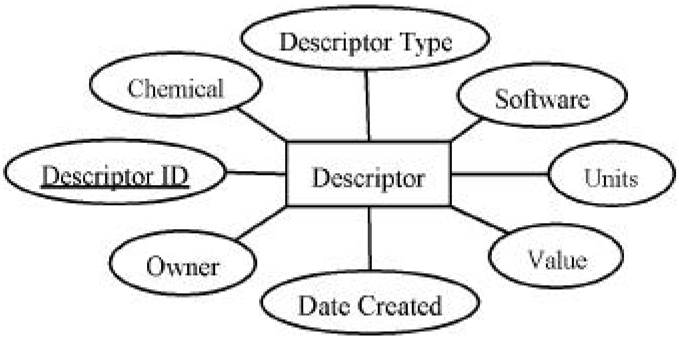


Figure 6: The Descriptor Entity

Descriptions of the attributes of this entity are described in ***Table 14***.

Table 14: Attributes of the Descriptor Entity

Attribute	Identifier	Description
Descriptor ID	Yes	Internal descriptor identifier
Chemical ID	No	The internal identifier associated with the chemical. A search must be carried out with the addition of each internal chemical to ensure that the current chemical is in fact a private chemical.
Descriptor Type	No	The type of descriptor calculated. This value for this attribute must be provided by the Descriptor Type entity.
Owner	No	The person, laboratory or data source that calculated the descriptor
Date Created	No	The date the calculation occurred, or if not found in the data source, the date this was entered into MOLDw
Software	No	The software that was used in the calculation
Value	No	The value of the descriptor
Units	No	The units associated with the descriptor

Within OpenMolGRID we are only concerned with single value descriptors. Descriptors that are multi valued or structured are not of concern according to the user requirements.

6.1.5 Property Type

All properties generated are of a particular type. To give flexibility to allow new types to be added, this entity has been identified. The attributes of this entity are shown in ***Figure 7***

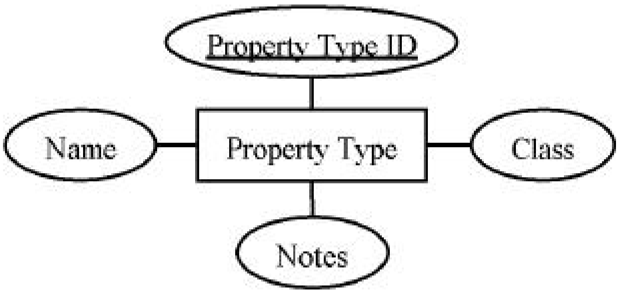


Figure 7: The Property Type Entity

Descriptions of the attributes of this entity are described in ***Table 15***.

Table 15: Attributes of the Property Type Entity

Attribute	Identifier	Description
Property Type ID	Yes	The internal identifier associated with the property type
Name	No	The name of the property.
Class	No	The class of the property.
Notes	No	Some other information about the property in textual format

6.1.6 Property

Properties will be produced as a result of experiments. The attributes of this entity are shown in ***Figure 8***.

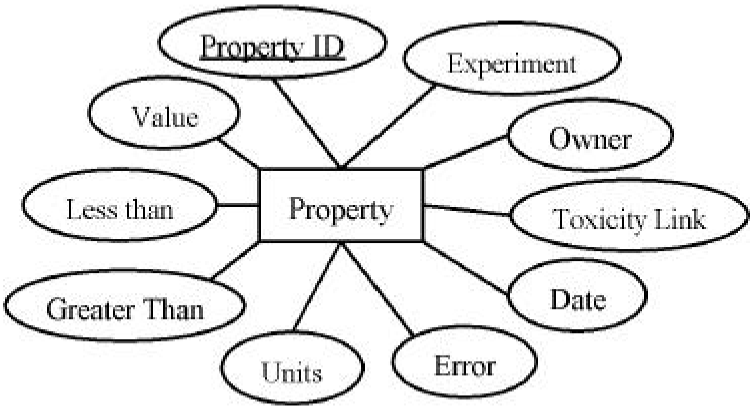


Figure 8: The Property Entity

Descriptions of the attributes of this entity are described in ***Table 16***.

Table 16: Attributes of the Property Entity

Attribute	Identifier	Description
Property ID	Yes	Internal identifier for the property
Experiment	No	The experiment in which the property was produced.
Value	No	The value of the property if a single value is possible
Less than	No	Can be used as part of a range of values
Greater than	No	If a value is supplied here then the value is a range
Units	No	The units associated with the property
Error	No	The error associated by the property
Owner	No	The individual, laboratory or source in which the property was calculated.
Date	No	The date the property was created. If not available in the original source, this value will be the date in which the value entered MOLDW.
Toxicity ID	No	If the property relates to toxicity other additional information may be available in the

6.1.7 Experiment

While the ultimate goal is to predict properties and descriptors without the need for the traditional experimental approach, the information relating to previous experiments is nonetheless important. An entity to represent the conditions surrounding the experiment is therefore required. The Experiment entity is shown in ***Figure 9***.

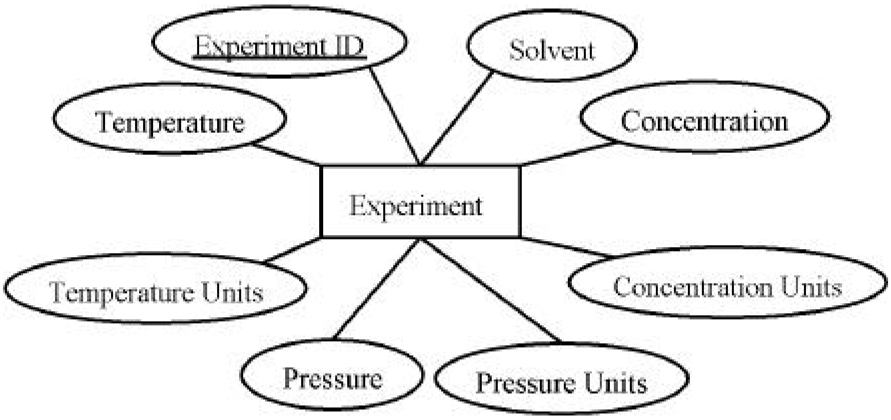


Figure 9: The Experiment Entity

Descriptions of the attributes of this entity are described in ***Table 17***.

Table 17: Attributes of the Experiment Entity

Attribute	Identifier	Description
Experiment ID	Yes	The internal ID associated with the experiment
Temperature	No	The temperature under which the experiment was carried out. This is important for some physicochemical properties: Specific Gravity, Density, Vapor Pressure, Solubility, Viscosity, Refractive Index
Temperature Units	No	The units associated with the temperature.
Pressure	No	The pressure under which the experiment was carried out. This is important for some physicochemical properties: Boiling Point
Pressure Units	No	The units associated with the pressure.
Concentration	No	The concentration under which the experiment was carried out. This is important for some physicochemical properties: Ph value
Concentration Units	No	The units associated with the concentration.
Solvent	No	The solvent used in the experiment. This is important for some physicochemical properties: Solubility

6.1.8 Toxicity Measure

While several properties may be stored within MOLDW, there is a specific requirement to store toxicity measures. The toxicity measure entity is shown in ***Figure 10***.

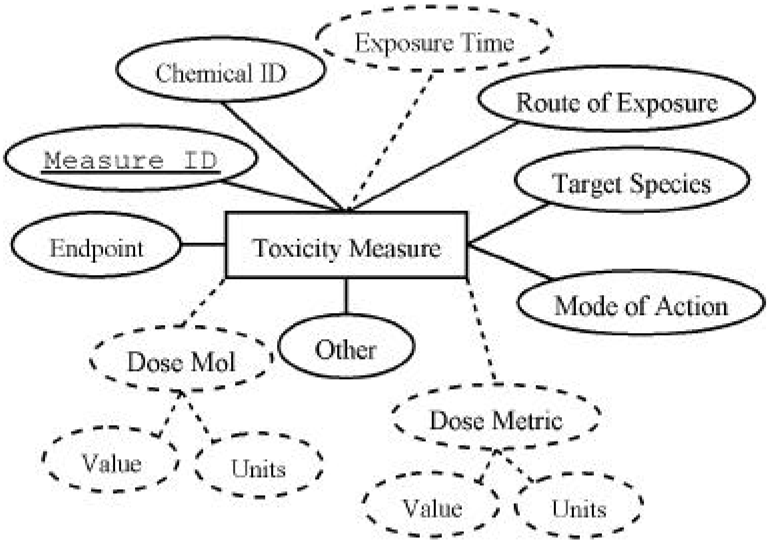


Figure 10: The Toxicity Measure Entity

Descriptions of the attributes of this entity are described in ***Table 18***

Table 18: Attributes of the Toxicity Measure Entity

Attribute	Identifier	Description
Measure ID	Yes	Internal identifier for the toxicity measure
Chemical ID	No	The ID of the chemical in question
Target Species	No	The name of the species the measure was carried out with.
Exposure Time	No	The amount of time the target was exposed to the chemical measured in hours.
Route of Exposure	No	The way in which the chemical was introduced to the target.
Mode of Action	No	The way in which the chemical affected the target.
Endpoint	No	The type of toxicity measure carried out e.g. LC_50, LD_50
Dose Mol value	No	The mol dosage used for the toxicity measure. This is a derived field and its transformation is described in 7.3
Dose Mol Units	No	The units of the mol dose. This will either be millimols per kilogram (mmol/kg) or millimols per litre (mmol/l)
Log Inverse	No	The log inverse of the Mol dosage. This is a derived field and its transformation is described in 7.5
Dose Metric Value	No	The metric dosage used in the toxicity measure. This is a derived field and its transformation is described in 7.4
Dose Metric Units	No	The units of the metric dose. This will either be milligrams per kilogram (mg/kg) or milligrams per litre (mg/l)

6.1.9 Software

For several different calculations there are various types of software that can be used. As the calculations are dependant on the software used, it is important to have a software entity as shown in ***Figure 11***.

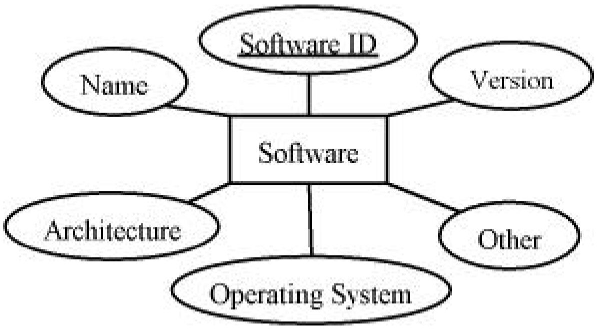


Figure 11: The Software Entity

Descriptions of the attributes of this entity are described in ***Table 19***.

Table 19: Attributes of the Software Entity

Attribute	Identifier	Description
Software ID	Yes	Internal ID used to identify the software
Name	No	The name of the software
Version	No	The version of the software
Architecture	No	The architecture of the system on which the software is running (if known)
Operating System	No	The operating system of the system on which the software is running (if known)
Other	No	Any other information that may be required

6.1.10 Owner

There are certain pieces of data contained within MOLDW that will have been generated by an individual, laboratory or that has come from a particular data source. It may be important for an individual to restrict results to one of these owners and so the owner entity is required. This entity is shown in ***Figure 12***.

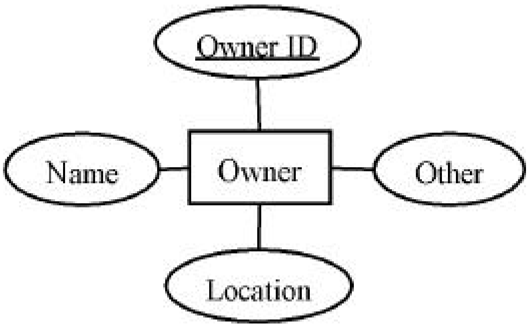


Figure 12: The Owner Entity

Descriptions of the attributes of this entity are described in ***Table 20***.

Table 20: Attributes of the Owner Entity

Attribute	Identifier	Description
Owner ID	Yes	Internal ID to represent an owner
Name	No	The name of the individual, laboratory or data source
Location	No	The location of the individual, laboratory or data source
Other	No	Other information associated with the owner.

6.1.11 Reference

Within many data sources, papers in which the toxicity measure was reported are often found. This type of information may be important and so the reference entity is required as shown in ***Figure 13***.

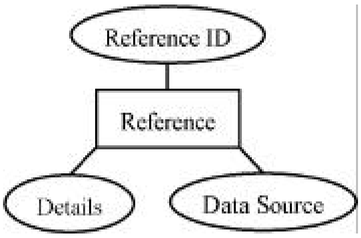


Figure 13: The Reference entity

Descriptions of the attributes of this entity are described in ***Table 21***.

Table 21: Attributes of the Reference Entity

Attribute	Identifier	Description
Reference ID	Yes	Internal ID to represent a reference.
Reference	No	The details associated with the reference.
Data Source	No	The data source from which the reference was found.

6.1.12 Measure References

The need for this entity comes from the fact that there existed a many to many relationship between Toxicity Measure and Reference. It is possible for the same reference to be referenced in multiple toxicity measurements.

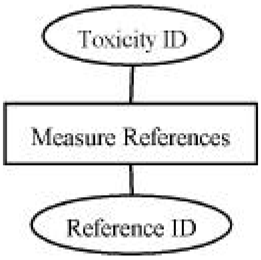


Figure 14: The Measure Reference Entity

Descriptions of the attributes of this entity are described in ***Table 22***.

Table 22: Attributes of the Measure Reference Entity

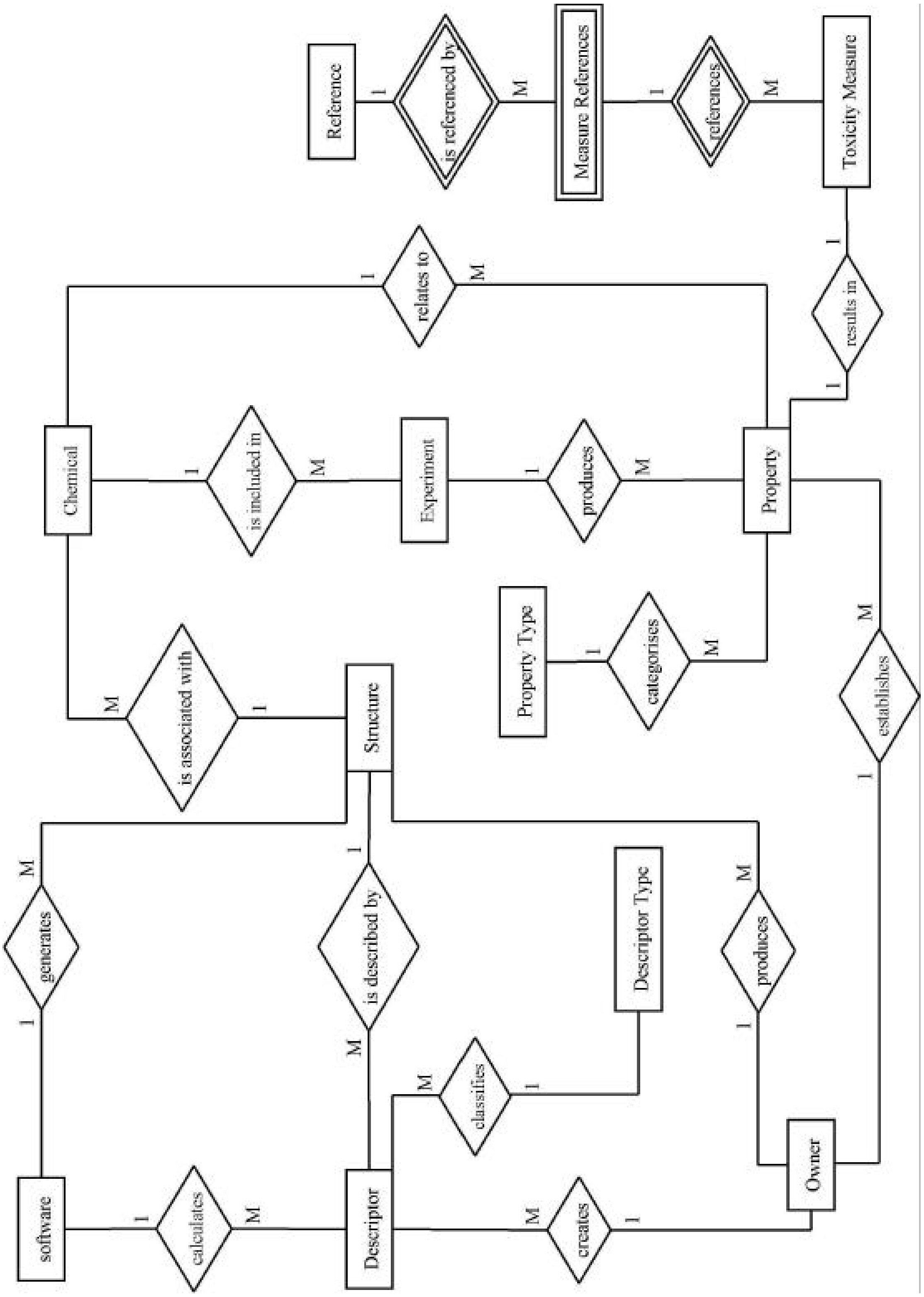
Attribute	Identifier	Description
Toxicity ID	Yes	Internal ID to represent a Toxicity Measure
Reference ID	Yes	Internal ID to represent a Reference

6.1.13 Relationships

Several relationships exist between the entities that have been identified. This section of the document will describe these relationships.

Entity From	Entity To	Cardinality	Description
Chemical	Property	1-M	A chemical identifies many properties
Property Type	Property	1-M	One property type categorises many properties
Chemical	Structure	1-M	A chemical is associated with many structures
Structure	Descriptor	1-M	One structure is described by many descriptors
Owner	Structure	1-M	One Owner develops many structures
Software	Structure	1-M	One Software generates many structures
Descriptor Type	Descriptor	1-M	One descriptor type classifies many descriptors
Owner	Descriptor	1-M	One Owner creates many Descriptors
Software	Descriptor	1-M	One Software calculates many Descriptors
Chemical	Experiment	1-M	One chemical is included in many experiments
Experiment	Property	1-M	One experiment produces many properties
Owner	Property	1-M	One owner establishes many properties
Property	Toxicity Measure	1-1	One Property results in one Toxicity Measure AND one Toxicity measure results in one Property.
Toxicity Measure	Measure References	1-M	One Toxicity Measure references many Measure References
Reference	Measure References	1-M	One Reference is referenced by many Measure References

6.2. Entity Relationship Diagram



7. Derived Fields and Transformations

This section of the document will describe the derived fields to be stored in MOLDW and in addition will describe the transformations that must take place in order to represent the data required in a form that will be useful to the end users of MOLDW. These transformations are simple, but during the evolution of MOLDW these will become more complicated.

7.1. Molecular Weight

7.1.1 Purpose

If the molecular weight is not found in a particular data source, then it can be calculated from the chemical formula. Chemical formula is usually provided in data sources. If a structure is available the Molecular weight can also be calculated from this.

7.1.2 Source

A chemical formula or Structure

7.1.3 Data Transformation

Using the CDK *MFAAnalyser* class the input of either a structure or formula can produce the molecular weight on invocation of the *getMass()* method. If a structure is available the molecular formula can also be calculated by invoking the *getMolecularFormula()* method.

7.1.4 Exception Handling

If neither molecular formula or structure are available then the field must be left blank.

7.1.5 Comments

None

7.2. Exposure Time

7.2.1 Purpose

The units for the exposure time will be hours

7.2.2 Source

The source of the information will be from the underlying data sources. If the units here are not expressed in hours, then data transformations will be required.

7.2.3 Data Transformation

Exposure units will be converted to hours. Therefore any unit that does not align will be converted to hours. The formula used will depend on the actual units.

Original Unit	Conversion Details	Precision
Minutes	Minutes/60	Two decimal places
Days	Num Days * 24	
Weeks	Num Weeks *7 *24	

7.2.4 Exception Handling

If not supplied then the field must be left blank. It will be the decision of the user to decide what this means.

7.2.5 Comment

Other formula may need to be specified at a later date.

7.3. Dose Mol

7.3.1 Purpose

The dose of a chemical measured in millimols per kilogram, or millimols per litre.

7.3.2 Source

Two items are needed: Molecular Weight from the Chemical identification field, and Dose Metric from the toxicity field.

7.3.3 Data Transformation

A simple transformation is required. Dose Mol is computed by the following formula:

$$\text{Dose Mol} = \frac{\text{Dose Metric/Mol. Weight}}{1000.}$$

7.3.4 Exception Handling

If Dose Metric or Mol. Weight are missing, leave the field blank.

7.3.5 Comment

The calculation returns a value which refers to the initial Dose Metric Units, so, if Dose Metric is expressed in milligrams/litre the output will be in millimols/litre, while grams/litre returns mols/litre.

7.4. Dose Metric

7.4.1 Purpose

The dose of a chemical measured in milligrams per kilogram, or milligrams per litre.

7.4.2 Sources

Two items are needed: Molecular Weight from the Chemical Identification field, and Dose Mol from the toxicity field.

7.4.3 Data Transformation

A simple transformation is required. Dose Metric is computed by the following formula:

$$\text{Dose Metric} = \frac{\text{Dose Mol * Mol. Weight}}{1000.}$$

7.4.4 Exception Handling

If Dose Mol or Molecular Weight are missing, leave the field blank.

7.4.5 Comment

The calculation returns a value which refers to the initial Dose Mol Units, so, if Dose Mol is expressed in millimols/litre the output will be in milligrams/litre, while mols/litre returns grams/litre. This is exactly the inverse transformation with respect the previous field.

7.5. Log Inverse

7.5.1 Purpose

To compute the logarithm to the base 10 of the inverse of Dose Mol expressed in mmol/l.

7.5.2 Sources

One item is needed: Dose Mol from the toxicity field.

7.5.3 Data Transformation

Log inverse (Dose Mol) is computed by the following formula: Log inverse = Log[1.0/Dose Mol/(mmol/l)]. Log inverse has no units.

7.5.4 Exception Handling

If Dose Mol equals zero, no mathematical meaning can be attributed to its inverse, therefore the field should be left blank.

7.5.5 Comment

The calculation returns a value that is often used to model a toxicological end-point such as LC50. It can be calculated only when Dose Mol is available, otherwise Dose Mol has to be computed before calculating the Log inverse transformation.

8. Standards

In order to help improve the quality of the final MOLDW a certain number of constraints must be followed. These are discussed below.

8.1. Programming Standards

As the development team within MOLDW contains multiple people it is important that all code be written in a similar format. This will help to maintain consistency between different parts of MOLDW. The standards that will be followed are referred to in the Project Quality Plan [15].

8.2. Document and Code Reviews

In order to ensure good software quality each document and code module will undergo a formal review process within the development environment of MOLDW in adherence to the quality standards outlined in the Project Quality Plan [15].

8.3. Source Control

It is important to maintain a list of changes to source code to help track where erroneous conditions have been introduced and to enable previous versions to be retrieved. The Concurrent Versions System (CVS) will be used to maintain a history of all versions of source code. This will also help to eradicate some common mistakes that can occur during the development of a multi-person project, such as accidental deletion of source files. This is in accordance with Project Quality Plan [15].

8.4. Error Logs

During program execution it is possible that errors will occur. Some of these errors may be fatal causing a system crash, while others may be more subtle and may take some time to discover. In either case it is important that the occurrence of such conditions is monitored. Error logs for use in MOLDW will provide a mechanism for logging such conditions. This has a distinct advantage in that it provides a source of information to the administrator of such a system to help them carry out their day-to-day duties.

8.5. Diagnostic Logs

The logging of system bugs is useful but does not in itself help solve a particular problem. It only notifies of the existence of a problem. A trace of the programs execution at the time of the bug's occurrence can provide a valuable insight into what is happening within a program and can thus help to solve problems. Diagnostic logs should therefore be used in the development of MOLDW.

8.6. Testing

Testing is an invaluable part of a software project. It is important that testing does not only happen during user acceptance testing. At all phases of the project testing should occur. This includes the unit testing and unit integration testing phases. During the development of MOLDW test plans will be written to formalise the testing process and a record of the success or otherwise of each test carried out will be maintained. This is in accordance with the project quality plan [15].

8.7. Bug Tracking

During testing, other phases of a project lifecycle or during day-to-day administration activities, problems or bugs will be discovered. Within MOLDW bugs will be tracked using a software tool called Bugzilla as outlined in the project Quality Plan [15]. This will have the advantage that a history of problems is maintained and these can be used as a reference if similar problems occur.

8.8. Security

Security of MOLDW will be provided using UNICORE's built in security mechanism as highlighted in D4.5a [4]. Although not required at this stage a users and privileges database is included in the high-level architecture (Figure 1) for future use. This may be used to further constrain access to different parts of MOLDW in the future, but is not currently required as all data has public access within the virtual organisation.

8.9. Reliability

MOLDW should be reliable and where it is not any errors should be traceable. This can be made possible with the use of error logs, diagnostic logs and bug tracking. To ensure this the quality procedures outlined in the project quality plan [15]will be followed.

8.10. Maintainability

MOLDW should be maintainable. It is therefore important that development occurs using the design constraints highlighted in section 6 of this document. If all code is written adhering to the programming standards, as described in the Project Quality Plan [15], then the developed code will all have the same look and feel. This means that subsequent developers who are adhering to the same or similar standards should be able to adapt their knowledge with relative ease. Documentation and commented code are of utmost importance.

8.11. Portability

Java will be used as the primary language for the software components DBAT_MOLDW and Populate MOLDW. This language has built-in portability. However, portability becomes an issue with the actual database implementation. MOLDW will be build using a relational database management system (RDBMS). Not all RDBMSs support SQL in the same way, so the physical implementation will differ depending on the choice of RDBMS. Within OpenMolGRID the choice of RDBMS was between MySQL and PostgreSQL. For reasons highlighted in the document technical DBMS Comparisons [14] PostgreSQL was chosen as the implementation RDBMS.

8.12. Extensibility

MOLDW will be designed using UML and will therefore use object-oriented concepts. It is currently envisaged that one small "program" will be written to integrate each data source into MOLDW in a similar way to the individual DBAT for access to different databases via UNICORE. In this way new parts can be added with relative ease by the addition of a new "program".

8.13. Reusability

MOLDW will be written using object-oriented concepts. Inherent in these concepts is the promotion of code reuse. Techniques such as inheritance will be used to help promote the reuse of certain components.

9. References

Some references in this section relate to documents stored on the BSCW server. These references all start with a location of "/OpenMolGRID". The BSCW server is located at https://hermes.chem.ut.ee/bscw/bscw.cgi.

[1] E. Benfenati and A. Papp, "Properties and priorities of the data for pharmaceutical and phytopharmaceutical compounds," /OpenMolGRID/Workpackage 1/ Deliverables/OpenMolGRID-12-D1.3a-0108-1-0, 15/09/03.

[2] S. Sild, "Description of the molecular engineering procedure," /OpenMolGRID/Workpackage 3/Deliverables/OpenMolGrid-3-D3.6-0103-1-0, 15/09/03.

[3] S. Sild, "Description of the quantitative structure property / activity relation model: model building and application," /OpenMolGRID/Workpackage 2/Deliverables/OpenMolGrid-2-D2.4a-0102-1-0, 15/09/03.

[4] M. Romberg and B. Schuller, "Description of the OpenMolGRID Grid architecture, security architecture, and infrastructure and the deployment of the project testbed," /OpenMoplGRID/Workpackage 4/Deliverables/OpenMolGRID-4-D4.5a-0104-1-0-architecture, 15/09/03.

[5] D. McCourt, "Description of Data Warehousing," /OpenMolGRID/Workpackage 1/ Deliverables/OpenMolGRID-1-D1.4e-0112-1-0-DescriptionofDataWarehousing, 15/09/03.

[6] D. McCourt, J. Jing and W. Dubitzky, "Description of Ecotox," /OpenMolGRID/Workpackage 1/ Deliverables/OpenMolGRID-1-D1.4a-0109-1-0, 15/09/03.

[7] D. McCourt, J. Jing and W. Dubitzky, "Description of NTP," /OpenMolGRID/Workpackage 1/ Deliverables/OpenMolGRID-1-D1.4b-0110-1-0, 12/09/03.

[8] D. McCourt, J. Jing and W. Dubitzky, "ECOTOX - Terretox data specification," /OpenMolGRID/Workpackage 1/ Deliverables/OpenMolGRID-1-D1.1b-0102-1-0-ECOTOX-TerretoxDataSpecification, 15/09/03.

[9] D. McCourt, J. Jing and W. Dubitzky, "ECOTOX - Aquire data specification," /OpenMolGRID/Workpackage 1/ Deliverables/OpenMolGRID-1-D1.1c-0103-1-0-ECOTOX-AquireDataSpecification, 15/09/03.

[10] D. McCourt, "NTP data specification," /OpenMolGRID/Workpackage 1/ Deliverables/OpenMolGRID-1-D1.1d-0104-1-0-NTPDataSpecification, 15/09/03.

[11] B. Schuller and M. Romberg, "Specification of Database Access Interface," /OpenMolGRID/Workpackage 4/ Deliverables/ OpenMolGRID-4-D4.1c-0103-2-0, 15/09/03.

[12] D. McCourt, "Specification of the Database Access Tool for the OpenMolGRID Data Warehouse," /OpenMolGRID/Workpackage 1/ Deliverables/OpenMolGRID-1-D1.1f-0106-2-0-DBAT_MOLDW, 15/09/03.

[13] M. Romberg and B. Schuller, "Specification of the generic user interface for database access," /OpenMolGRID/Workpackage 4/ Deliverables/OpenMolGRID-4-D4.1a-0101-2-0, 15/09/03.

[14] J. Jing and D. McCourt, "DBMS Comparisons," /OpenMolGRID/Workpackage 1/Technical Documents/ OpenMolGRID-1-TED-0113-2-0 15/09/03.

[15] G.H.F. Diercksen, "Project Quality Plan," /OpenMolGRID/Workpackage 7/Deliverables/OpenMolGRID-7-D7.1-0101-1-2-ProjectQualityPlan, 12/09/03.